

"README.DOC" File  
Release Notes for MICROSOFT(R) QuickBASIC  
Version 4.0 for IBM(R) Personal Computers  
and Compatibles

(C) Copyright Microsoft Corporation, 1987

THIS FILE CONTAINS IMPORTANT INFORMATION CONCERNING VERSION 4.0 OF MICROSOFT(R) QuickBASIC. PLEASE READ THE ENTIRE FILE BEFORE USING QuickBASIC.

This file has seven parts:

PART    CONTENTS

- 1      Information about additions and changes to the Learning and Using Microsoft QuickBASIC manual.
- 2      Information about additions and changes to the BASIC Language Reference manual
- 3      Information about additions and changes to the Programming in BASIC: Selected Topics manual
- 4      Using your Mouse with QuickBASIC
- 5      Using QuickBASIC with 3.5-inch floppy disks
- 6      Supplementary information on mixed-language programming
- 7      Using Btrieve with QuickBASIC

===< Part 1: Changes for Learning and Using Microsoft QuickBASIC >=====

Page    Correction

-----

---    Some keyboards have an extra set of DIRECTION (i.e. arrow) keys, in addition to those on the numeric keypad. A bug in the ROM BIOS of some machines with these keyboards can interfere with the QuickBASIC editor.    Disk 3 (disk 2, if you are using 3.5-inch disks) includes a program, FIXSHIFT.COM, that fixes this bug. If you have such a keyboard, run this program by typing FIXSHIFT. If your machine does not have the bug, FIXSHIFT displays a message telling you so. Otherwise FIXSHIFT prompts you for the appropriate actions. FIXSHIFT takes about 450 bytes of memory. Except for the BIOS bug, it has no effect on other programs you run.

---    QuickBASIC Version 4.0 supports Hercules (R) display adapters. See the entry for the SCREEN statement in Part 2, below.

xxii   If you install an IBM (R) Personal System/2 (TM) Video Graphics Array display adapter (VGA) in a non-PS/2 machine, the VGA adapter should be the only adapter in the system, and you should not use mono modes (SCREEN 10) if you have a color monitor. Similarly, you should not use color modes (SCREEN 1, 2, 7, 8, 9, 11, 12, 13) if you have a monochrome monitor.

11    Suggestions for using QuickBASIC with a single-floppy system:

Disk 1  
=====

QB.EXE  
Your program source file  
(Optional: Operating system files)  
(Optional: QB.QLB)

Disk 2  
=====  
BC.EXE  
LINK.EXE  
LIB.EXE  
BRUN40.EXE  
BRUN40.LIB

Disk 3  
=====  
BCOM40.EXE

When you make executable files and Quick libraries from within the environment, you have to swap the disks in and out of your disk drive to make the proper programs and libraries available. When specific files cannot be found on the currently inserted disk, you will see the prompt "Cannot find file <filename>." Before you switch disks, type B: and press ENTER. (You can do this because, even though you do not have a physical drive B, DOS recognizes a logical drive B.) Insert the appropriate disk when prompted. Note that when the linker prompts for a path, you must also include the name of the file for which it is searching.

- 12 See Part 4 of this file for additional information on installing and using your Mouse with QuickBASIC.

-- There is a new option to LINK.EXE

Option: /NOE[XTDICTIONARY]

If the linker suspects that a public symbol has been redefined, it prompts you to link again with the /NOE option. When you do so, it searches the individual object files, rather than "dictionaries" it has created, to resolve conflicts. For example, when linking a program with NOEM.OBJ or NOCOM.OBJ, you must use the /NOE option.

-- NOEM.OBJ is a supplied object file. It allows stand-alone executable files compiled with the /O option to be substantially smaller when run on machines equipped with a math coprocessor (8087 or 80287). Note that programs linked with NOEM.OBJ do not run on machines that do not have a math coprocessor. If all your stand-alone executable files are always run on machines with coprocessors, you can use the following method to change the library that contains floating-point emulation routines (BCOM40.LIB) to produce smaller executable files:

LIB BCOM40.LIB -QB4EM.OBJ+NOEM.OBJ;

This command replaces the emulator math support in the stand-alone run-time library with support specific to the coprocessor.

Alternatively, you can explicitly link NOEM.OBJ from the command line on a case-by-case basis. For example, to create PROG.EXE using NOEM.OBJ, compile PROG.BAS with the /O option (either from within QuickBASIC or from the command line), then link as follows:

LINK PROG.OBJ+NOEM.OBJ/NOE;

245 PTR86 is no longer supported. Use VARSEG and VARPTR instead.  
Also, when used with a communications device LOF now returns the  
amount of space remaining (in bytes) in the output buffer. In  
previous versions this was returned in the input buffer. Also, note  
that a variable and SUB procedure could have the same name in  
previous versions. In Version 4.0, this causes a "duplicate definition" error.

===< Part 2: Changes for BASIC Language Reference >=====

Page Correction

-----

18 The program FLPT.BAS, as printed in the manual, produces a run-time  
error. The program has been corrected in the on-disk version.  
See the \SOURCE directory on disk 3 (or disk 2 if you are using  
3.5 inch disks).

84 The description of CALL ABSOLUTE says the file ABSOLUTE.OBJ must be  
linked with the program. This file is no longer supplied. Link with  
QB.LIB instead.

122 String concatenation is not permitted in a CONST statement.

373 The SCREEN statement now includes mode 3 for Hercules display  
adapters. The following is a brief summary of screen mode 3.  
See your Hercules documentation for details.

- QuickBASIC supports Hercules Graphics Card, Graphics  
Card Plus, InColor Card, and 100% compatibles
- You must use a monochrome monitor
- Hercules text mode is SCREEN 0; Hercules Graphics mode  
is SCREEN 3.
- You must load the Hercules driver (QBHERC.COM) before running  
your program. If the driver is not loaded, SCREEN 3 statement  
gives an "Illegal function call" error message. Type QBHERC  
to load the driver.
- Text dimensions are 80x25 (9x14 character box); bottom  
2 scan lines of 25th row are not visible.
- Resolution is 720x348 pixels, monochrome.
- Number of screen pages supported is 2.
- The PALETTE statement is not supported.
- In order to use the Mouse, you must follow special  
instructions for Hercules cards in the Microsoft Mouse  
Programmer's Reference Guide. (This must be ordered  
separately; it is not supplied with either the QuickBASIC  
or the Mouse package.)

477 - The "Advanced feature unavailable" error message may occur  
if you are using DOS Version 2.1 when trying to use a fea-  
ture supported only in later versions (i.e. file locking).

484 - The "Duplicate definition" error message also occurs if you  
have a SUB or FUNCTION procedure with the same name as a

variable. In previous versions, this did not cause an error.

513 - Message 2024 ("Name : symbol multiply defined, use /NOE")  
should be numbered 2044

-----< New or Revised Error Messages >-----

Error Number	Explanation
--------------	-------------

L1003	New message: "/QUICKLIB, /EXEPACK incompatible" You specified both options, /QUICKLIB and /EXEPACK, but these two options cannot be used together.
L2024	Message should read: "<name> : symbol already defined" Explanation: The linker has found a public-symbol redefinition. Remove extra definition(s).
L2043	Message should read: "Quick library support module missing" You did not specify, or LINK could not find, the object module or library required for creating a Quick library. In the case of QuickBASIC, the library provided is BQLB40.LIB
L4003	This message should be deleted from documentation.
U4157	LIB warning message: "Insufficient memory, extended dictionary not created."
U4158	LIB warning message: "Internal error, extended dictionary not created." Both of these LIB warnings just indicate that LIB was unable to create the extended dictionary. The library is still valid, but the linker cannot take advantage of extended dictionaries to link faster.

===< Part 3: Changes for Programming in BASIC: Selected Topics >=====

141 In the "Information Returned" column for the LOC function, the  
description should read: "The amount of space remaining (in  
bytes) in the output buffer". (Note that this is a change from  
the behavior of LOF in previous versions of QuickBASIC).

===< Part 4: Using your Mouse with QuickBASIC >=====

-----< New Mouse Driver for Use with QuickBASIC >-----

QuickBASIC Version 4.0 can be used with any mouse that is 100%  
compatible with the Microsoft Mouse. However, you must use a  
Microsoft Mouse driver Version 6.00 or later. Earlier versions may  
cause unpredictable behavior when used with QuickBASIC. MOUSE.COM,  
Version 6.11 is supplied with QuickBASIC Version 4.0.

Especially if you are writing programs that use the mouse, you  
should use the supplied version of the mouse driver when working in  
QuickBASIC. Previous versions have included MOUSE.SYS, which is  
installed by including the line DEVICE=MOUSE.SYS in your CONFIG.SYS  
file. This version of QuickBASIC includes MOUSE.COM, which is not  
installed via CONFIG.SYS. To install MOUSE.COM, just type MOUSE at  
the DOS prompt. To include MOUSE.COM automatically when your  
machine boots, make sure MOUSE.COM is in your search path, then put the  
line

## MOUSE

in your AUTOEXEC.BAT file. To free up memory, you can remove the mouse driver at any time by typing MOUSE OFF at the DOS prompt. This will restore between 9K and 10.5K of memory with Version 6.11.

### -----< Using Mouse Function Calls from QuickBASIC Programs >-----

If you are programming for the Microsoft Mouse, you should obtain the Microsoft Mouse Programmer's Reference Guide and the library MOUSE.LIB that comes with it. (These are not included in QuickBASIC or Mouse package and must be ordered separately). Most of the information in the Mouse programmer's reference guide applies directly to QuickBASIC Version 4. However, the following additional restrictions must be observed:

Certain Mouse function calls (Functions 9 & 16) require you to set up an integer array and pass the address of the array to the mouse driver. For previous versions, the only restriction on this array was that it had to be \$STATIC (the default array type). In QuickBASIC Version 4.0, however, the array also must be in a COMMON block if you will be making the Mouse function call from within the QuickBASIC environment. In addition, it is recommended that the support code for the Mouse call be in a Quick library or linked into the executable file when making Mouse function calls from QuickBASIC.

To produce a Quick library for using Mouse function calls from within the QuickBASIC environment, use the following command line (produces MOUSE.QLB):

```
LINK MOUSE.LIB/QU,MOUSE.QLB,,BQLB40.LIB/NOE;
```

An example from PIANO.BAS (included with the Microsoft Mouse Programmer's Reference) for using Mouse function call 9:

```
DEFINT A-Z
DECLARE SUB MOUSE (M1, M2, M3, M4)
DIM Cursor(15, 1)
COMMON Cursor() ' ensures array data is in DGROUP
.
. (set up Cursor() for mouse cursor shape desired)
.
M1 = 9: M2 = 6: M3 = 0
CALL MOUSE(M1, M2, M3, VARPTR(Cursor(0, 0)))
```

In addition to the above, note that Mouse function calls 21-23 require dynamically allocated storage out of the home data segment. The recommended way to do this is to allocate space in a dynamic string variable based on the return value from function call 21, using the STRING\$ or SPACE\$ function, and using VARPTR on this string variable just prior to calling Mouse function call 22 or 23.

### ====< Part 5: Using QuickBASIC with 3.5-inch Floppy Disks >=====

If you have two 3.5-inch floppy-disk drives

Copy the files from the distribution disks to three other disks, as follows:

Disk 1: All files from distribution disk 1, plus all BRUN40 and BQLB files appearing on distribution disk 2

Disk 2: BCOM40.LIB, BC.EXE, LINK.EXE, LIB.EXE

Disk 3: BASIC source files, Quick libraries, and QB.QLB

Most of the time you can just use disks 1 and 3, with disk 1 in drive A and disk 3 in drive B. When you make a Quick library or an executable file to run from DOS, you will see the prompt:

Cannot find file <filename>

Insert disk 2 in drive A and press ENTER. If the program requires a Quick library, make sure both the Quick library and the corresponding stand-alone (.LIB) library are on the disk in drive B.

If you have only a single 3.5-inch floppy disk drive

Prepare disks 1, 2, and 3 as with a 2-drive system. However, when you see the prompt "Cannot find file <filename>", type B: and press ENTER before inserting the proper disk. (You can do this because, even though you do not have a physical drive B, DOS recognizes a logical drive B.) Insert the appropriate disk when prompted. Note that when the linker prompts for a path, you must also include the name of the file for which it is searching.

====< Part 6: Supplementary Information on Mixed-Language Programming >=====

-----< Linking from within QuickC or with QCL >-----

Microsoft QuickC and the QCL command both set the /NOI linker by default. Therefore, you should not link from within QuickC, or with QCL, when your program contains modules written in a case-insensitive language such as BASIC. Use LINK to link your program from the command line.

-----< Pascal and Fortran Modules in QuickBASIC Programs >-----

Modules compiled with Microsoft Pascal or Fortran can be linked with BASIC programs, as described in the Microsoft Mixed-Language Programming Guide. They can also be incorporated in Quick libraries. However, QuickBASIC programs containing code compiled with Microsoft Pascal must allocate at least 2K near-heap space for Pascal. This can be done by using the DIM statement to allocate a static array of 2K or greater in the NMALLOC named common block, for example, as follows:

```
DIM name%(2048)
COMMON SHARED /NMALLOC/ name%()
```

The Pascal runtime assumes it always has at least 2K of near-heap space available. If the Pascal code cannot allocate the required space, QuickBASIC may crash. This applies to Pascal code in Quick libraries as well as Pascal code linked into executable files. The situation is similar for Fortran I/O, which also requires near buffer space, and which can be provided by the same means as the Pascal near malloc space.

-----< STATIC Array Allocation >-----

If you are writing assembly-language modules for use in QuickBASIC programs, see Section 2.3.3, "Variable Storage Allocation," in the BASIC Language Reference. Assembly-language code should not assume

data is in a particular segment. To avoid problems, pass data using the SEG or CALLS keywords, or use FAR pointers. Alternatively, you can declare all arrays dynamic (still using far pointers) since dynamic arrays are handled identically by BC and withinQuickBASIC.

-----< Quick Libraries with Leading Zeros in the First Code Segment >--

A Quick library containing leading zeros in the first CODE segment is invalid, causing the message "Error in loading file <name> - Invalid format" when you try to load it in QuickBASIC. For example, this can occur if an assembly-language routine puts data that is initialized to zero in the first CODE segment, and it is subsequently listed first on the LINK command line when you make a Quick library. If you have this problem, do either of the following:  
(1) link with a BASIC module first on the LINK command line, or  
(2) make sure that, in whatever module comes first on the LINK command line, the first code segment starts with a non-zero byte.

===< Part 7: Using Btrieve with QuickBASIC >=====

If you use Btrieve with QuickBASIC, you must make a small change to your programs for QuickBASIC Version 4.0. Currently your programs contain a statement that obtains the address of the field buffer for an open file. For example:

```
OPEN "NUL" AS #1
FIELD #1, 20 AS CITY$, 10 AS STATE$
FCB.ADDR% = VARPTR(#1)           'This statement obtains
                                the address
```

In QuickBASIC Version 4.0, you should change the indicated statement to return the address of the first variable in your field buffer minus a constant, as follows:

```
OPEN "NUL" AS #1
FIELD #1, 20 AS CITY$, 10 AS STATE$
FCB.ADDR% = SADD(CITY$) - 188     'CITY$ is the first field
                                buffer      variable
```

Your programs should function correctly with Btrieve with this change.